UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

| APPLICATION NO. | FILING DATE | FIRST NAMED INVENTOR | ATTORNEY DOCKET NO. | CONFIRMATION NO. |
|---|---|---|---|---|
| 10/823,282 | 04/12/2004 | John Erik Lindholm | NVDA P000735 | 4571 |

7590 01/29/2007

MOSER PATTERSON & SHERIDAN LLP
Suite 100
595 Shrewsbury Avenue
Shrewsbury, NJ 07702

| EXAMINER | |
|---|---|
| HSU, JONI | |
| **ART UNIT** | **PAPER NUMBER** |
| 2628 | |

| SHORTENED STATUTORY PERIOD OF RESPONSE | MAIL DATE | DELIVERY MODE |
|---|---|---|
| 3 MONTHS | 01/29/2007 | PAPER |

**Please find below and/or attached an Office communication concerning this application or proceeding.**

If NO period for reply is specified above, the maximum statutory period will apply and will expire 6 MONTHS from the mailing date of this communication.

| Office Action Summary | Application No. | Applicant(s) | |
|---|---|---|---|
| | 10/823,282 | LINDHOLM ET AL. | |
| | Examiner | Art Unit | |
| | Joni Hsu | 2628 | |

*-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --*

**Period for Reply**

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

**Status**

1)☐ Responsive to communication(s) filed on _____.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

**Disposition of Claims**

4)☒ Claim(s) 1-25 is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) 1-25 is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

**Application Papers**

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

**Priority under 35 U.S.C. § 119**

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

**Attachment(s)**

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

## DETAILED ACTION

### *Claim Rejections - 35 USC § 101*

1. 35 U.S.C. 101 reads as follows:

> Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

2. Claims 1-16 are rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter.

Claims 1 and 9 are directed to an abstract idea of processing divergent graphics samples in a programmable processing unit, rather than a practical application of the abstract idea. The claimed invention as a whole must accomplish a practical application. That is, it must produce a "useful, concrete and tangible result (*State Street*, 149 F.3d at 1373, 47 USPQ2d at 1601-02). The tangible requirement requires that the claim must set forth a practical application of the 101 judicial exception to produce a real-world result (*Benson*, 409 U.S. at 71-72, 175 USPQ at 676-77). See MPEP 2106 II A. Since there is no tangible result recited in these claims, these claims are directed to non-statutory subject matter.

Claims 2-8 and 10-16 are non-statutory for the same reasons discussed above.

*Claim Rejections - 35 USC § 103*

3. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

4. The factual inquiries set forth in *Graham* **v.** *John Deere Co.*, 383 U.S. 1, 148 USPQ 459 (1966), that are applied for establishing a background for determining obviousness under 35 U.S.C. 103(a) are summarized as follows:

1. Determining the scope and contents of the prior art.
2. Ascertaining the differences between the prior art and the claims at issue.
3. Resolving the level of ordinary skill in the pertinent art.
4. Considering objective evidence present in the application indicating obviousness or nonobviousness.

5. Claims 1-18 and 22-25 are rejected under 35 U.S.C. 103(a) as being unpatentable over Guttag (US006173394B1) in view of Lueh (US006966057B2).

6. With regard to Claim 1, Guttag discloses a method for processing divergent graphics samples in a programmable graphics processing unit (*graphics processor 71 controls the fetching and branching operations*, Col. 13, lines 53-57), the method comprising incrementing a subroutine depth of a first sample to designate that first call instructions are to be executed on the first sample (*if the following instruction word is at the next sequential address, program control flow unit 130 post increments the program counter*, Col. 15, lines 58-67; *during normal*

*sequential instruction operation program flow control unit 130 increments bit 3 of program counter PC 701 to address the next 64 bit instruction, writing to program counter PC 701 executes a subroutine call*, Col. 89, lines 35-41); and executing the first call instructions on the first sample (Col. 17, lines 43-52; Col. 16, lines 12-16).

However, Guttag does not teach pushing state data of a second sample upon which the first call instructions are not to be executed onto a global stack. However, Lueh discloses pushing state data of a second sample upon which the first call instructions are not to be executed onto a global stack (*saving the live global state includes pushing the live global state onto a stack, and restoring the saved live global state includes retrieving it from the stack*, Col. 9, lines 4-35).

It would have been obvious to one of ordinary skill in the art at the time of invention by applicant to modify the device of Guttag to include pushing state data of a second sample upon which the first call instructions are not to be executed onto a global stack as suggested by Lueh. Lueh discloses that executing a field watch sequence involves this pushing of state data onto a global stack, and pushing state data onto a global stack prevents the call instructions from being executed (Col. 9, lines 22-33). This prevents significant runtime overhead caused by un-activated call instructions because it guards the execution of the instruction (Col. 7, lines 9-16). Therefore, pushing state data onto a global stack guards call instructions that are not needed from being executed, and therefore prevents significant runtime overhead caused by un-activated call instructions.

7. With regard to Claim 2, Guttag discloses the step of holding the second sample idle (*delay of two instructions is experienced before any branch takes effect*, Col. 16, lines 22-30).

8. With regard to Claim 3, Guttag does not teach that holding the second sample idle comprises encoding the second sample with non-operation information. However, Lueh discloses that holding the second sample idle comprises encoding the second sample with non-operation information (*replace the jump instruction with a no-op sequence*, Col. 9, lines 4-35).

It would have been obvious to one of ordinary skill in the art at the time of invention by applicant to modify the device of Guttag so that holding the second sample idle comprises encoding the second sample with non-operation information as suggested by Lueh. Lueh discloses that by encoding the sample with non-operation information, this implements the guard to prevent execution of the code (Col. 9, lines 4-13). This prevents significant runtime overhead caused by un-activated call instructions because it guards the execution of the instruction (Col. 7, lines 9-16). Therefore, encoding the sample with non-operation information guards call instructions that are not needed from being executed, and therefore prevents significant runtime overhead caused by un-activated call instructions.

9. With regard to Claim 4, Guttag discloses the step of determining whether the first call instructions include a call return that contains second call instructions (*instruction 1 sets the zero "Z" status bit if D1=D2, instruction 2 is conditional based upon the zero "Z" status bit, if the zero "Z" status bit is "0", then instruction 2 is not performed, if the zero "Z" status bit is "1",*

*then instruction 2 is performed, subroutine return is a conditional operation*, Col. 161, lines 20-36).

10. With regard to Claim 5, Guttag discloses the step of incrementing the subroutine depth of the first sample to designate that second call instructions (next loop, software branch) are to be executed on the first sample (*If the next instruction is at the next sequential address, program control flow unit 130 post increments program counter PC 701 during the fetch pipeline state. Note this post increment means that program counter PC 701 stores the address of the next instruction to be fetched. Otherwise, program control flow unit 130 loads the address of the next instruction into program counter PC 701 according to loop logic 720 or software branch*, Col. 93, lines 21-33).

11. With regard to Claim 6, Guttag discloses the step of executing the second call instructions (next loop, software branch) on the first sample (Col. 93, lines 21-33; Col. 17, lines 43-52; Col. 16, lines 12-16).

12. With regard to Claim 7, Guttag does not teach that pushing state data removes the second sample from a working set of data. However, Lueh discloses the guard prevents execution of the sample, or removes the sample from a working set of data, by saving the state data, which includes pushing state data onto a stack. The sample is restored to the working set of data by restoring the saved state data, which includes retrieving it from the stack (Col. 9, lines 4-35). Therefore pushing state data onto a stack means that pushing state data removes the second

sample from a working set of data, and the second sample is restored to the working set of data by retrieving it from the stack. This would be obvious for the same reasons given in the rejection for Claim 1.

13. With regard to Claim 8, Guttag discloses the step of determining whether an instruction in an instruction sequence includes a call-return that contains the first call instructions (*if the count stored in loop count register LC0 is non-zero, the program flow returns to loop start address 2 that is the same as loop start address 0 which repeats the loop starting with instruction block 1, if the count stored in loop count register LC0 is "0", the program ends*, Col. 96, lines 57-62).

14. With regard to Claim 9, Guttag discloses a method of processing divergent graphics samples in a programmable graphics processing unit (Col. 13, lines 53-57), the method comprising identifying a first sample (two instructions) having a first subroutine depth (two); holding idle a second sample (branch) having a second subroutine depth (one), the first subroutine depth being greater than the second subroutine depth; executing first return instructions on the first sample; and popping state data of a second sample from a delay slot position (*a delay of two instructions is experienced before any branch takes effect, the pipelined operation requires this delay, since the next two instruction following such a branch instruction have already been fetched, instructions may be placed in the two delay slot positions*, Col. 16, lines 22-30).

However, Guttag does not teach popping state data of a second sample from a global stack. However, Lueh discloses popping state data of a second sample from a global stack (Col. 9, lines 4-35). This would be obvious for the same reasons given in the rejection for Claim 1.

15. With regard to Claim 10, Claim 10 is similar in scope to Claim 3, and therefore is rejected under the same rationale.

16. With regard to Claim 11, Guttag does not teach that popping the state data of the second sample restores the second sample to a working set of data. However, Lueh discloses that popping the state data of the second sample restores the second sample to a working set of data (Col. 9, lines 4-35). This would be obvious for the same reasons given in the rejection for Claim 1.

17. With regard to Claim 12, Guttag discloses the step of decrementing the first subroutine depth, making the first subroutine depth equal to the second subroutine depth (*with loop count register LS2 equal to zero, and loop count register LC1 not equal to zero: loop count register LC1, the outer loop, is decremented, loop count register LC2 is reloaded with the value of loop reload register LR2, which is zero; program counter 701 is loaded with the address stored in loop start register LS1, which is the address of the one instruction loop, thus the instruction repeats, the loop ends by the loop count register LC1 decrementing to zero, in this case the program continues with instruction 8, the next following instruction*, Col. 163, lines 15-34).

18. With regard to Claim 13, Guttag discloses the step of determining whether the first subroutine depth is greater than zero (*loop count register LC1 not equal to zero,* Col. 163, lines 15-34).

19. With regard to Claim 14, Guttag discloses the step of identifying in an instruction sequence a next instruction to be executed if the first subroutine depth is equal to zero (*loop count register LC1 may decrement to zero, in this case the program continues with instruction 8, the next following instruction,* Col. 163, lines 26-34).

20. With regard to Claim 15, Guttag discloses the step of executing second return instructions on the first sample and the second sample if the first subroutine depth is greater than zero (*loop count register LC1 not equal to zero, thus the instruction repeats,* Col. 163, lines 15-34).

21. With regard to Claim 16, Guttag discloses the step of decrementing the first subroutine depth and the second subroutine depth (Col. 163, lines 15-34).

22. With regard to Claim 17, Guttag discloses a system for processing divergent graphics samples in a programmable graphics processing unit (Col. 13, lines 53-57), the system comprising a subroutine depth scoreboard (registers LC1 and LC2) configured to store a first subroutine depth corresponding to a first sample and a second subroutine depth corresponding to a second sample; and a remap configured to increment and decrement the first subroutine depth and the second subroutine depth in the subroutine depth scoreboard (Col. 163, lines 15-34).

However, Guttag does not teach a global stack configured to store state data; and a remap configured to push state data onto and to pop state data from the global stack. However, Lueh discloses a global stack configured to store state data; and a remap configured to push state data onto and to pop state data from the global stack (Col. 9, lines 4-35). This would be obvious for the same reasons given in the rejection for Claim 1.

23. With regard to Claim 18, Guttag discloses incrementing the first subroutine depth to designate that the first call instructions are to be executed on the first sample, and does not execute the second sample (loop, software branch) until the first sample is finished (*if the following instruction word is at the next sequential address, program control flow unit 130 post increments the program counter, otherwise, program control flow unit 130 loads the address of the next instruction word according to the loop logic or software branch*, Col. 15, lines 58-67). Therefore, the remap is further configured to determine that first call instructions are to be executed on the first sample, but not the second sample, to increment the first subroutine depth to designate that the first call instructions are to be executed on the first sample

However, Guttag does not teach pushing state data of the second sample onto the global stack. However, Lueh discloses pushing state data of the second sample onto the global stack (Col. 9, lines 4-35). This would be obvious for the same reasons given in the rejection for Claim 1.

24. With regard to Claim 22, Guttag discloses a system for processing divergent graphics samples in a programmable graphics processing unit (Col. 13, lines 53-57), the system

comprising means for incrementing a first subroutine depth of a first sample to designate that first call instructions are to be executed on the first sample (Col. 15, lines 58-67); means for executing the first call instructions on the first sample (Col. 17, lines 43-52; Col. 16, lines 12-16); means for identifying that the first subroutine depth is greater than a second subroutine depth of the second sample; means for executing first return instructions on the first sample; (Col. 16, lines 22-30); means for decrementing the first subroutine depth (Col. 163, lines 15-34); and means for popping state data of the second sample from the delay slot positions (Col. 16, lines 22-30).

However, Guttag does not teach means for pushing state data of a second sample upon which the first call instruction are not to be executed onto a global stack; and means for popping state data of the second sample from the global stack. However, Lueh discloses means for pushing state data of a second sample upon which the first call instruction are not to be executed onto a global stack; and means for popping state data of the second sample from the global stack (Col. 9, lines 4-35). This would be obvious for the same reasons given in the rejection for Claim 1.

25. With regard to Claims 23-25, these claims are similar in scope to Claims 2, 3, and 15 respectively, and therefore are rejected under the same rationale.

26. Claims 19-21 are rejected under 35 U.S.C. 103(a) as being unpatentable over Guttag US006173394B1) and Lueh (US006966057B2) in view of Gossett (US006236413B1).

27. With regard to Claim 19, Guttag and Lueh are relied upon for the teachings as discussed above relative to Claim 18. Guttag discloses that the address of the next instruction word to be fetched is stored in program counter PC 701. Theses instruction words are for a digital image/graphics processor 71. Writing to program counter PC 701 executes a subroutine call. The write alters program counter PC 701 (*The major task of program flow control unit 130 is control of instruction fetch during the fetch pipeline state. The address of the next instruction word to be fetched is stored in program counter PC 701, instruction word of digital image/graphics processor 71. During normal sequential instruction operation program flow control unit 130 increments bit 3 of program counter PC 701 to address the next 64 bit instruction. Writing to program counter PC 701 executes a subroutine call. The write alters program counter PC 701. Writing to a different register address designated branch BR executes a software branch*, Col. 89, lines 22-50). The address of the next instruction word to be fetched that is stored in program counter PC is considered to be a codeword. Writing this codeword to program counter PC 701 programmably configures the digital image/graphics processor to execute the subroutine call. According to the disclosure of this application, dispatching a PC token into the computation units means that the codewords included in the PC token configure the various computation units to perform the operations specified in the one or more instructions corresponding to the codewords [0035]. Guttag discloses that writing the codeword to the program counter PC 701 executes the subroutine call in the digital image/graphics processor 71, which means that the codeword in the program counter PC 701 configures the digital image/graphics processor 71 to perform the operations specified in the subroutine call corresponding to the codeword. Therefore, Guttag discloses that the remap is further configured

to generate a PC token for executing the first call instructions, the PC token containing one or more codewords that configure programmable computation units to execute the first call instructions, and to dispatch the PC token into the computation units, followed by the first sample and the second sample (*The major task of program flow control unit 130 is control of instruction fetch during the fetch pipeline state. The address of the next instruction word to be fetched is stored in program counter PC 701, instruction word of digital image/graphics processor 71. During normal sequential instruction operation program flow control unit 130 increments bit 3 of program counter PC 701 to address the next 64 bit instruction. Writing to program counter PC 701 executes a subroutine call. The write alters program counter PC 701. Writing to a different register address designated branch BR executes a software branch*, Col. 89, lines 22-50).

However, Guttag does not teach that the remap is further configured to encode the second sample with non-operation data. However, Lueh discloses that the remap is further configured to encode the second sample with non-operation data (Col. 9, lines 4-35). This would be obvious for the same reasons given in the rejection for Claim 1.

However, Guttag and Lueh do not teach that the programmable computation units are within a recirculating shader pipeline. However, Gossett discloses generating a token for executing the first call instructions, the tokens configuring programmable computations units within a recirculating shader pipeline (*multi-pass processing is implemented through the use of recirculation pipes built into the components of the graphics processing subsystem, texture-shader subsystem*, Col. 3, lines 5-16) to execute the first call instructions, and to dispatch the token into the recirculating shader pipeline, followed by the first sample (vertex) and the second

sample (*multi-pass operation is started by sending a MP_Begin token to the texture-shader subsystem 133, this is followed by one or more command tokens that set enables and modes, each pass is indicated by a vertex token, at the end of all passes, an MP_End token is sent, this entire program is repeated for each vertex*, Col. 12, lines 5-14).

It would have been obvious to one of ordinary skill in the art at the time of invention by applicant to modify the devices of Guttag and Lueh so that the programmable computation units are within a recirculating shader pipeline as suggested by Gossett because Gossett suggests the advantage of servicing the bandwidth requirements of graphics subsystems efficiently, without a large amount of redundant logic, while retaining the ability to perform complex graphics operations to create a realistic 3D image (Col. 2, lines 29-40; Col. 1, lines 42-44).

28. With regard to Claim 20, Guttag discloses that the remap is further configured to determine that first return instructions are to be executed on the first sample, but not the second sample (Col. 16, lines 22-30), remap is further configured to generate a PC token for executing the first return instructions, the PC token containing one or more codewords that configure programmable computation units to execute the first return instructions, and to dispatch the PC token into the computation units, followed by the first sample and the second sample (Col. 89, lines 22-50).

However, Guttag does not teach that the remap is further configured to encode the second sample with non-operation data. However, Lueh discloses that the remap is further configured to encode the second sample with non-operation data (Col. 9, lines 4-35). This would be obvious for the same reasons given in the rejection for Claim 1.

However, Guttag and Lueh do not teach that the programmable computation units are within a recirculating shader pipeline. However, Gossett discloses generating a token for executing the first return instructions, the tokens configuring programmable computations units within a recirculating shader pipeline (Col. 3, lines 5-16) to execute the first return instructions, and to dispatch the token into the recirculating shader pipeline, followed by the first sample and the second sample (Col. 12, lines 5-14). This would be obvious for the same reasons given in the rejection for Claim 19.

29. With regard to Claim 21, Guttag discloses that the remap is further configured to decrement the first subroutine depth and to pop state data of the second sample (Col. 163, lines 15-34).

However, Guttag does not teach popping state data of the second sample from the global stack. However, Lueh discloses popping state data of the second sample from the global stack (Col. 9, lines 4-35). This would be obvious for the same reasons given in the rejection for Claim 1.

***Prior Art of Record***

The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

Hussain (US006252610B1) teaches a multipass rendering system including a graphics pipeline organized as a sequential series of tasks. The host computer passes a first token through the graphics pipeline. The first token causes the pipeline tasks to select their respective first state

information blocks. The host computer follows the first token by sending a selected group of one or more graphics primitives through the graphics pipeline. The graphics primitives are rendered by the pipeline tasks, using the state information included in the first state information blocks. The host processor then sends a second token through the graphics pipeline. The second token causes the pipeline tasks to select their respective second state information blocks (Col. 2, lines 25-35, 44-67).

***Conclusion***

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Joni Hsu whose telephone number is 571-272-7785. The examiner can normally be reached on M-F 8am-5pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Ulka Chauhan can be reached on 571-272-7782. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

JH

ULKA CHAUHAN
SUPERVISORY PATENT EXAMINER